

The image is a large, symmetrical, abstract graphic composed of the letters 'S' and 'Y' arranged in a grid-like pattern. The overall shape is a stylized 'Y' or a complex letterform. The top part is a wide horizontal bar made of 'S's, with 'Y's forming a central vertical stem. The sides of the 'Y' are also formed by 'S's and 'Y's, creating a sense of depth and structure. The letters are black on a white background, and the arrangement is highly regular and repetitive, suggesting a digital or algorithmic origin.

(1)	90	DECLARATIONS
(1)	114	EXESSUSPND - SUSPEND SYSTEM SERVICE
(2)	179	KERNEL AST THAT SUSPENDS PROCESS
(2)	231	EXESRESUME - RESUME SYSTEM SERVICE
(2)	276	EXESHIBER - HIBERNATE SYSTEM SERVICE
(2)	326	EXESWAKE - WAKE SYSTEM SERVICE
(2)	387	EXESNAMPID - CONVERT PROCESS NAME TO PID
(2)	516	EXESxPID TO xxx - CONVERT PID TO OTHER PID OR PCB ADDRESS
(2)	650	EXESSETPRN - SET PROCESS NAME

```
0000 1 .TITLE SYSPCNTRL PROCESS CONTROL SERVICES
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7 *
0000 8 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 9 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 10 * ALL RIGHTS RESERVED.
0000 11 *
0000 12 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 13 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 14 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 15 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 16 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 17 * TRANSFERRED.
0000 18 *
0000 19 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 20 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 21 * CORPORATION.
0000 22 *
0000 23 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 24 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 25 *
0000 26 *****
0000 27
0000 28 ++
0000 29 FACILITY: EXECUTIVE, PROCESS CONTROL SYSTEM SERVICES
0000 30
0000 31 ABSTRACT:
0000 32 THIS MODULE CONTAINS THE ROUTINES WHICH IMPLEMENT THE PROCESS
0000 33 CONTROL SERVICES, SUSPEND, RESUME, HIBERNATE AND WAKE.
0000 34
0000 35 AUTHOR:
0000 36 R. HUSTVEDT
0000 37
0000 38 MODIFIED BY:
0000 39
0000 40 V03-013 LJK0256 Lawrence J. Kenah 7-Dec-1983
0000 41 Only allow ASTs if XQP thread is active. Clear SUSPEN bit
0000 42 if pool allocation fails.
0000 43
0000 44 V03-012 CWH3012 CW Hobbs 27-Sep-1983
0000 45 In EXESIPID_TO_EPID treat a null IPID as a special case,
0000 46 and return the null.
0000 47
0000 48 V03-011 LJK0250 Lawrence J. Kenah 31-Aug-1983
0000 49 Set the SUSPEN bit before lowering IPL to zero to insure
0000 50 that the PCB of the target process has not disappeared.
0000 51
0000 52 Make the SUSPND AST a regular kernel AST so that it properly
0000 53 interlocks with the XQP. Include the interlocking code.
0000 54
0000 55 V03-010 CWH1007 CW Hobbs 14-May-1983
0000 56 Enable the storing of the actual cluster node info in the
0000 57 high bits of the EPID.
```

```
0000 58 :
0000 59 :
0000 60 :
0000 61 :
0000 62 :
0000 63 :
0000 64 :
0000 65 :
0000 66 :
0000 67 :
0000 68 :
0000 69 :
0000 70 :
0000 71 :
0000 72 :
0000 73 :
0000 74 :
0000 75 :
0000 76 :
0000 77 :
0000 78 :
0000 79 :
0000 80 :
0000 81 :
0000 82 :
0000 83 :
0000 84 :
0000 85 :
0000 86 :
0000 87 :
0000 88 :---
```

V03-009 CWH1003 CW Hobbs 27-Apr-1983
Change pid conversion routines to do more checking
of pid against pids stored in PCB. Make sure that
condition codes reflect final value in R0.

V03-008 LJK0197 Lawrence J. Kenah 25-Mar-1983
Insure that all success paths raise IPL to SYNCH.

V03-007 ACG0321 Andrew C. Goldstein, 24-Mar-1983 0:19
Allow non-privileged control over processes of equal UIC

V03-006 ROW0168 Ralph O. Weber 3-MAR-1983
Change W^ references to G^.

V03-005 CWH1002 CW Hobbs 19-Feb-1982
Modify EXESNAMPID to use extended PIDs, add PID conversion
routines:

- EXESIPID_TO_PCB - internal pid to pcb address
- EXESEPID_TO_PCB - extended pid to pcb address
- EXESIPID_TO_EPID - internal pid to extended pid
- EXESEPID_TO_IPID - extended pid to internal pid

V03-004 LJK0188 Lawrence J. Kenah 22-Oct-1982
Do not allow processes that are being deleted to also
be suspended.

V03-003 KDM46395 Kathleen D. Morse 28-Jun-1982
Change word displacement to longword.

```
0000 90      .SBTTL  DECLARATIONS
0000 91
0000 92      :
0000 93      : INCLUDE FILES:
0000 94      :
0000 95
0000 96      $ACBDEF      : DEFINE AST CONTROL BLOCK
0000 97      $IPLDEF      : IPL DEFINITIONS
0000 98      $PCBDEF      : PCB OFFSET DEFINITIONS
0000 99      $PRDEF       : PROCESSOR REGISTER DEFS
0000 100     $PRIDEF      : PRIORITY INCREMENT DEFINITIONS
0000 101     $PRVDEF      : PRIVILEGE BIT DEFINITIONS
0000 102     $RSNDEF      : RESOURCE NUMBER DEFINITIONS
0000 103     $SSDEF       : STATUS DEFINITIONS
0000 104     $STATEDEF    : SCHEDULER STATE DEFINITIONS
0000 105     :
0000 106     : EQUATED SYMBOLS:
0000 107     :
00000004 0000 108     PID=4      : DISPLACEMENT TO PID ARGUMENT
00000008 0000 109     PRCNAM=8   : DISPLACEMENT TO PROCESS NAME
0000 110
00000000 111     .PSECT  AEXENONPAGED,BYTE  : NONPAGED EXEC
0000 112
```

```
0000 114 .SBTTL EXE$SUSPND - SUSPEND SYSTEM SERVICE
0000 115 :++
0000 116 EXE$SUSPND - SUSPEND SYSTEM SERVICE
0000 117
0000 118 FUNCTIONAL DESCRIPTION:
0000 119 EXE$SUSPND IMPLEMENTS THE SUSPEND PROCESS SYSTEM SERVICE.
0000 120 THIS SERVICE CAUSES THE SPECIFIED PROCESS TO BE SUSPENDED
0000 121 BY INITIATING A KERNEL MODE AST IF NOT THE CURRENT PROCESS.
0000 122 A SUSPENDED PROCESS CANNOT RECEIVE ASTS AND WILL ONLY BE
0000 123 RESUMED AS A RESULT OF THE RESUME SYSTEM SERVICE OR A
0000 124 DELETE PROCESS REQUEST.
0000 125
0000 126
0000 127 CALLING SEQUENCE:
0000 128 CALLG  ARGLIST,EXE$SUSPND
0000 129
0000 130
0000 131 INPUT PARAMETERS:
0000 132 04(AP) - PROCESS IDENTIFICATION POINTER (PID)
0000 133 08(AP) - PROCESS NAME DESCRIPTOR POINTER
0000 134 R4 - PCB ADDRESS OF CURRENT PROCESS
0000 135
0000 136 IMPLICIT INPUTS:
0000 137 PCB OF CURRENT PROCESS
0000 138 PCB OF TARGET PROCESS
0000 139
0000 140
0000 141 OUTPUT PARAMETERS:
0000 142 R0 - COMPLETION STATUS
0000 143 @PID(AP) - PROCESS IDENTIFICATION OF TARGET PROCESS
0000 144
0000 145 COMPLETION CODES:
0000 146 $$$_NORMAL - NORMAL SUCCESSFUL COMPLETION
0000 147 $$$_NOPRIV - INSUFFICIENT PRIVILEGE FOR REQUESTED OPERATION
0000 148 $$$_NONEXPR - NON-EXISTENT PROCESS
0000 149 $$$_ACCVIO - ACCESS VIOLATION ON WRITE DESTINATION
0000 150 $$$_INSFMEM - INSUFFICIENT DYNAMIC MEMORY FOR REQUEST
0000 151 ( ONLY RETURNED IF NO RESOURCE WAIT ENABLE )
0000 152
0000 153 SIDE EFFECTS:
0000 154 NONE
0000 155
0000 156 --
0000 157
0000 158 EXE$SUSPND::
0000 159 .WORD ^M<R2,R3,R4,R5> ; SUSPEND SYSTEM SERVICE
0002 160 BSBW EXE$NAMPID ; REGISTER SAVE MASK FOR R2-R5
0005 161 BLBC R0,30$ ; TRANSLATE AND VERIFY ARGS
0008 162 PUSHL R1 ; CONTINUE IF NO ERROR
000A 163 BBS #PCBSV_DELPEN,PCBSL_STS(R4),20$ ; SAVE PID
000F 164 BBSS #PCBSV_SUSPEN,PCBSL_STS(R4),10$ ; EXIT IF BEING DELETED
0014 165 SETIPL #IPL$_ASTDEL ; ... OR IF ALREADY SUSPENDED
0017 166 BSBW EXE$ACLOCIRP ; ENABLE
001A 167 BLBC R0,EXIT_NO_POOL ; ALLOCATE I/O PACKET FOR AST
001D 168 MOVL R2,R5 ; IF LBC THEN NO PACKET ALLOCATED
0020 169 MOVAL B^SUSPND,ACBSL_AST(R5) ; SETUP POINTER TO AST CONTROL BLK
0025 170 CLRB ACBSB_RMOD(R5) ; SET FOR KERNEL AST ON PROCESS
; SET ACCESS MODE FOR AST
```

003C 0000
00AD 30 0002
30 50 E9 0005
51 DD 0008
24 24 A4 01 E0 000A
1D 24 A4 0B E2 000F
FFE6' 30 0017
30 50 E9 001A
55 52 D0 001D
10 A5 3A'AF DE 0020
0B A5 94 0025

OC	A5	8E	D0	0028	171	MOVL	(SP)+,ACB\$\$_PID(R5)	:	SET PID FOR AST
		52	D4	002C	172	CLRL	R2	:	SET NULL PRIORITY INCREMENT
		FFCF'	30	002E	173	BSBW	SCH\$QAST	:	QUEUE KERNEL AST
		78	11	0031	174 10\$:	BRB	EXITN	:	EXIT WITH NORMAL STATUS
				0033	175				
50	08E8	8F	3C	0033	176 20\$:	MOVZWL	#SS\$_NONEXPR,R0	:	RETURN 'NO SUCH PROCESS' IF DELPEN
		74	11	0038	177 30\$:	BRB	EXIT	:	ERROR RETURN

	003A	179				.SUBTITLE	KERNEL AST THAT SUSPENDS PROCESS
	003A	180					
	003A	181					
	003A	182				KERNEL AST ROUTINE TO SUSPEND PROCESS	
	003A	183					
	003A	184				CALLING SEQUENCE:	
	003A	185				(SAME EFFECT AS) DCLAST ASTADR=DELETE MODE=KERNEL	
	003A	186					
	003A	187				INPUT PARAMETERS:	
	003A	188				NONE	
	003A	189					
	003A	190				OUTPUT PARAMETERS:	
	003A	191				NONE	
	003A	192					
	003A	193				IMPLICIT INPUTS:	
	003A	194				PCB OF CURRENT PROCESS LOCATED VIA SCH\$GL_CURPCB	
	003A	195					
	003A	196				IMPLICIT OUTPUTS:	
	003A	197				PCB\$V_SUSPEN - CLEARED	
	003A	198					> WHEN PROCESS IS RESUMED
	003A	199				PCB\$V_RESPEN - CLEARED	
	003A	200					
	003A	201					
	003A	202				.ENABLE	LOCAL_BLOCK
	003A	203					
	003A	204				SUSPND:	
	003A	205				.WORD	*M<R2,R3,R4,R5>
	003C	206				MOVL	G^SCH\$GL_CURPCB,R4
	0043	207					
	0043	208				10\$:	MOVPSL -(SP)
	0045	209					SETIPL #IPL\$ SYNCH
	0048	210					BBCC #PCB\$V_RESPEN,PCB\$L_STS(R4),30\$; BR IF NO PENDING RESUME
	004D	211				EXIT_NO-	POOL:
	004D	212					BBCC #PCB\$V_SUSPEN,PCB\$L_STS(R4),20\$; CLEAR SUSPEND PENDING
	0052	213				20\$:	SETIPL #0
	0055	214					RET
	0056	215					
	0056	216				30\$:	TSTB PCB\$B_DPC(R4)
	0059	217					BEQL 40\$
	005B	218					BICB2 #1,PCB\$B_ASTACT(R4)
	005F	219					BSBW SCH\$NEWLVL
	0062	220					MOVL #RSN\$ ASTWAIT,R0
	0065	221					BSBW SCH\$RWAIT
	0068	222					BRB 10\$
	006A	223					
	006A	224				40\$:	MOVAL G^SCH\$GQ_SUSP,R2
	0071	225					BSBW SCH\$WAITR
	0074	226					BRB 10\$
	0076	227					
	0076	228				.DISABLE	LOCAL_BLOCK
	0076	229					
54	00000000'GF	003C	D0				
	7E	DC					
09 24 A4	05	E5					
00 24 A4	0B	E5					
	04						
	2A A4	95					
	0F	13					
0C A4	01	8A					
	FF9E'	30					
50	01	D0					
	FF98'	30					
	D9	11					
52	00000000'GF	DE					
	FF8C'	30					
	CD	11					

SYS	Sym
ACB	ACB
ACB	ACB
ACC	ACC
EVT	EVT
EXE	EXE
EXE	EXE
EXE	EXE
EXE	EXE
EXE	EXE
EXE	EXE
EXE	EXE
EXE	EXE
EXE	EXE
EXE	EXE
EXE	EXE
EXI	EXI
EXI	EXI
EXI	EXI
GOT	GOT
GOT	GOT
IPL	IPL
IPL	IPL
IVL	IVL
NEX	NEX
NOD	NOD
NON	NON
NOP	NOP
PCB	PCB
PCB	PCB
PCB	PCB
PCB	PCB
PCB	PCB
PCB	PCB
PCB	PCB
PCB	PCB
PCB	PCB
PCB	PCB
PCB	PCB
PCB	PCB
PCB	PCB
PCB	PCB
PCB	PCB
PCB	PCB
PCB	PCB
PCB	PCB
PCB	PCB
PID	PID
PIX	PIX
PR\$	PR\$
PRC	PRC
PRI	PRI
PRV	PRV
PRV	PRV
RET	RET
RSN	RSN

```
0076 231 .SBTTL EXES$RESUME - RESUME SYSTEM SERVICE
0076 232 :++
0076 233 EXES$RESUME - RESUME SYSTEM SERVICE
0076 234
0076 235 FUNCTIONAL DESCRIPTION:
0076 236 EXES$RESUME IMPLEMENTS THE RESUME SYSTEM SERVICE WHICH RESTARTS
0076 237 A SUSPENDED PROCESS.
0076 238
0076 239 INPUT PARAMETERS:
0076 240 R0(AP) - PROCESS IDENTIFICATION POINTER (PID)
0076 241 R8(AP) - PROCESS NAME DESCRIPTOR POINTER
0076 242 R4 - PCB ADDRESS OF CURRENT PROCESS
0076 243
0076 244 IMPLICIT INPUTS:
0076 245 PCB OF CURRENT PROCESS
0076 246 PCB OF TARGET PROCESS
0076 247 PROCESS HEADER OF CURRENT PROCESS
0076 248
0076 249 OUTPUT PARAMETERS:
0076 250 R0 - COMPLETION STATUS
0076 251 @PID - PROCESS IDENTIFICATION OF TARGET PROCESS
0076 252
0076 253 IMPLICIT OUTPUTS:
0076 254 NONE
0076 255
0076 256 COMPLETION CODES:
0076 257 $$$_NORMAL - NORMAL SUCCESSFUL COMPLETION
0076 258 $$$_NOPRIV - INSUFFICIENT PRIVILEGE FOR REQUESTED OPERATION
0076 259 $$$_NONEXPR - NON-EXISTENT PROCESS
0076 260 $$$_ACCVIO - ACCESS VIOLATION ON WRITE DESTINATION
0076 261
0076 262 SIDE EFFECTS:
0076 263 NONE
0076 264 :--
0076 265
0076 266 EXES$RESUME::
0076 267 .WORD *M<R2,R3,R4> ; RESUME SYSTEM SERVICE
0078 268 BSBB EXES$NAMPID ; REGISTER SAVE MASK FOR R2-R4
007A 269 BLBC R0,EXIT ; CONVERT AND VALIDATE
007D 270 MOVZBL #PRI$ RESAVL,R2 ; EXIT IF ERROR OCCURRED
0080 271 BBSS #PCB$V_RESPEN,PCB$L_STS(R4),10$ ; SET PRIORITY INCREMENT CLASS
0085 272 10$: RPTEVT RESUME ; SET RESUME PENDING
0089 273 BRB ; REPORT RESUME EVENT
008B 274 ; AND TAKE NORMAL EXIT
```

00 24 A4 31 52 02 05 20 11 001C 10 E9 9A E2 11

```
008B 276 .SBTTL EX$HIBER - HIBERNATE SYSTEM SERVICE
008B 277 :++
008B 278 EX$HIBER - HIBERNATE SYSTEM SERVICE
008B 279
008B 280 FUNCTIONAL DESCRIPTION:
008B 281 EX$HIBER IMPLEMENTS THE HIBERNATE SYSTEM SERVICE WHICH
008B 282 PLACES THE PROCESS IN A WAIT STATE, HIB, UNTIL IT
008B 283 IS RE-AWAKENED BY A WAKE SYSTEM SERVICE. ASTS MAY BE DELIVERED
008B 284 WHILE THE PROCESS IS IN A HIBERNATE STATE.
008B 285
008B 286
008B 287
008B 288 CALLING SEQUENCE:
008B 289 CALLG ARGLIST,EX$HIBER
008B 290
008B 291
008B 292 INPUT PARAMETERS:
008B 293 R4 - PCB ADDRESS OF CURRENT PROCESS
008B 294
008B 295 IMPLICIT INPUTS:
008B 296 PROCESS CONTROL BLOCK(PCB) OF THE PROCESS ISSUING THE HIBERNATE
008B 297 SYSTEM SERVICE.
008B 298
008B 299
008B 300 OUTPUT PARAMETERS:
008B 301 R0 - COMPLETION STATUS CODE
008B 302
008B 303 IMPLICIT OUTPUTS:
008B 304 NONE
008B 305
008B 306 COMPLETION CODES:
008B 307 $$$_NORMAL - NORMAL SUCCESSFUL COMPLETION
008B 308
008B 309 SIDE EFFECTS:
008B 310 THE PROCESS WILL BE PLACED IN A WAIT STATE UNTIL EITHER
008B 311 AN AST IS DELIVERED OR A WAKE REQUEST IS MADE.
008B 312
008B 313 :--
008B 314
008B 315
008B 316 EX$HIBER::
008B 317 .WORD *M<R2,R3,R4> ; HIBERNATE SYSTEM SERVICE
008B 318 SETIPL #IPL$ SYNCH ; REGISTER SAVE MASK FOR R2-R4
008B 319 BBCCI #PCB$V_WAKEPEN,PCB$L_STS(R4),10$ ; BLOCK SCHEDULING EVENTS
008B 320 BRB EXITN ; CHECK FOR PENDING WAKE
008B 321 ; AND RETURN TO CALLER
008B 322
008B 323 10$:
008B 324 MOVAL G^SCH$GQ_HIBWQ,R2 ; MUST HIBERNATE
BRW SCH$WAIT ; SET ADDRESS OF WAIT QUEUE HDR
; AND WAIT
```

001C
02 24 A4 0C E7
14 11
52 00000000'GF DE
FF5F' 31 009E

```
00A1 326 .SBTTL EXESWAKE - WAKE SYSTEM SERVICE
00A1 327 :++
00A1 328 EXESWAKE - WAKE SYSTEM SERVICE
00A1 329
00A1 330 FUNCTIONAL DESCRIPTION:
00A1 331 THE WAKE SYSTEM SERVICE CAUSES A PROCESS IN A HIBERNATE STATE
00A1 332 TO BE CHANGED TO AN EXECUTABLE STATE AND RE-EXECUTED.
00A1 333 IF THE TARGET OF A WAKE SERVICE IS NOT CURRENTLY HIBERNATING,
00A1 334 THEN A BIT IS POSTED WHICH WILL CAUSE A SUBSEQUENT HIBERNATE
00A1 335 CALL BY THAT PROCESS TO RETURN IMMEDIATELY.
00A1 336
00A1 337 CALLING SEQUENCE:
00A1 338 CALLG  ARGLIST,EXESWAKE
00A1 339
00A1 340 INPUT PARAMETERS:
00A1 341 04(AP) = PROCESS IDENTIFICATION (PID) OF PROCESS TO WAKE
00A1 342 08(AP) = ADDRESS OF PROCESS NAME DESCRIPTOR
00A1 343 R4 - PCB ADDRESS
00A1 344
00A1 345 IMPLICIT INPUTS:
00A1 346 PCB OF CURRENT PROCESS
00A1 347 ALL PCBs LOCATED BY THE VECTOR @SCH$GL_PCBVEC
00A1 348
00A1 349 OUTPUT PARAMETERS:
00A1 350 R0 - COMPLETION STATUS CODE
00A1 351 @PID(AP) - PROCESS IDENTIFICATION (PID) OF PROCESS AWAKENED
00A1 352
00A1 353 IMPLICIT OUTPUTS:
00A1 354 PCB$V WAKEPEN BIT IN PCB$L STS OF TARGET PROCESS WILL BE
00A1 355 SET IF PROCESS IS NOT HIBERNATING.
00A1 356
00A1 357 COMPLETION CODES:
00A1 358 $$$_NORMAL - NORMAL SUCCESSFUL COMPLETION
00A1 359 $$$_NONEXPR - NON-EXISTENT PROCESS
00A1 360 $$$_NOPRIV - NO PRIVILEGE FOR ATTEMPTED OPERATION
00A1 361 $$$_ACCVIO - ACCESS VIOLATION ON WRITE DESTINATION
00A1 362
00A1 363 SIDE EFFECTS:
00A1 364 THE TARGET PROCESS WILL BE CHANGED TO AN EXECUTABLE STATE,
00A1 365 COM OR COMO, IF IT IS IN A HIBERNATE STATE AND
00A1 366 RESCHEDULING WILL BE INITIATED IF NECESSARY.
00A1 367
00A1 368 :--
00A1 369 EXESWAKE::
00A1 370 .WORD  ^M<R2,R3,R4>          : WAKE SYSTEM SERVICE
00A1 371 BSBB   EXES$NAMPID          : SAVE MASK FOR R2-R4
00A1 372 :                               : CONVERT NAME TO PID
00A1 373 :
00A1 374 R0 - SUCCESS INDICATOR
00A1 375 R1 - PID CORRESPONDING TO NAME STRING
00A1 376 R4 - PCB ADDRESS IF NAME WAS FOUND
00A1 377
00A1 378 BLBC   R0,EXIT              : CONTINUE IF PROCESS LOCATED
00A1 379 BSBW   SCH$WAKE            : WAKE PROCESS BY PID
00A1 380 EXITN:  MOVZWL  #$$$_NORMAL,R0 : EXIT HIBERNATE SERVICE
00A1 381 EXIT:   : SET NORMAL COMPLETION
00A1 382 SETIPL #0                  : RETURN WITH R0 SET
                                : ENABLE
```

001C
0D 10

06 50 E9
FF55 30

50 01 3C

SYSPCNTRL
V04-000

PROCESS CONTROL SERVICES
EXESWAKE - WAKE SYSTEM SERVICE

D 6

16-SEP-1984 02:25:01 VAX/VMS Macro V04-00
5-SEP-1984 03:56:04 [SYS.SRC]SYSPCNTRL.MAR;1

Page 10
(2)

04 00B1 383 RET
00B2 384
00B2 385

; AND RETURN TO CALLER

SYS
V04

```
00B2 387 .SBTTL EXESNAMPID - CONVERT PROCESS NAME TO PID
00B2 388 ++
00B2 389 EXESNAMPID - CONVERT PROCESS NAME TO PID
00B2 390
00B2 391 FUNCTIONAL DESCRIPTION:
00B2 392 EXESNAMPID OBTAINS THE PROPER PID AND PCB ADDRESS FOR A
00B2 393 STANDARD PROCESS CONTROL SERVICE ARGUMENT LIST CONSISTING
00B2 394 OF A PID/PROCESS-NAME PAIR. THE ABSENCE OF BOTH SELECTS THE
00B2 395 CURRENT PROCESS. AFTER ANY NECESSARY NAME TRANSLATION AND
00B2 396 PID VALIDATION, GROUP AND WORLD PROCESS CONTROL PRIVILEGES
00B2 397 ARE CHECKED.
00B2 398
00B2 399
00B2 400 CALLING SEQUENCE:
00B2 401 JSB/BSB EXESNAMPID
00B2 402
00B2 403 INPUT PARAMETERS:
00B2 404 PID(AP) - ADDRESS OF PID SOURCE/DESTINATION (EXTENDED PID)
00B2 405 PRCNAM(AP) - POINTER TO PROCESS DESCRIPTOR TO CONVERT TO PID
00B2 406 R4 - PCB ADDRESS
00B2 407
00B2 408 IMPLICIT INPUTS:
00B2 409 @SCH$GL PCBVEC - VECTOR OF PCB ADDRESSES
00B2 410 PHD$$_PRIV - PRIVILEGE BIT VECTOR IN PROCESS HEADER
00B2 411
00B2 412 OUTPUT PARAMETERS:
00B2 413 R0 - COMPLETION STATUS
00B2 414 R1 - INTERNAL PROCESS IDENTIFICATION (PID) OF NAMED PROCESS.
00B2 415 ZERO IF NO MATCH IS FOUND.
00B2 416 R4 - PCB ADDRESS OF PROCESS IF MATCH IS FOUND.
00B2 417 @PID(AP) - EXTENDED PROCESS IDENTIFICATION (EPID) OF SELECTED PROCESS
00B2 418 IPL - IPL$_SYNCH (IPL UNCHANGED IF $$$_ACCVIO OR $$$_IVLOGNAM)
00B2 419
00B2 420 COMPLETION CODES:
00B2 421 $$$_NORMAL - NORMAL SUCCESSFUL COMPLETION
00B2 422 $$$_IVLOGNAM - INVALID LOGICAL NAME STRING
00B2 423 $$$_NONEXPR - NONEXISTENT PROCESS OR INVALID PID
00B2 424 $$$_NOPRIV - NO PRIVILEGE FOR SPECIFIED OPERATION.
00B2 425 $$$_ACCVIO - ACCESS VIOLATION FOR WRITE DESTINATION
00B2 426
00B2 427 SIDE EFFECTS:
00B2 428 NONE
00B2 429
00B2 430 --
00B2 431 EXESNAMPID::
00B2 432 MOVL PID(AP),R0
00B2 433 BEQL 10$,R0
00B2 434 IFNOWRT #4,(R0),ACCVIO
00B2 435 MOVL (R0),R1
00B2 436 BEQL 10$,R1
00B2 437 MOVL R1,R0
00B2 438 BSBW EX$EPID_TO_IPID
00B2 439 MOVL R0,R1
00B2 440 CLRL R0
00B2 441 BRB GOTPID
00B2 442 10$: MOVL PCB$_PID(R4),R1
00B2 443 MOVL PRCNAM(AP),R3
00B2 444

: TRANSLATE PNAME TO PID
: GET PID ADDRESS
: NO PID ADDRESS
: ERROR IF ACCESS VIOLATION
: NOW FETCH (EXTENDED) PID
: BRANCH IF NO PID FOUND
: PASS EPID TO ROUTINE IN R0
: CONVERT TO IPID
: NOW R1 HAS THE USEFUL IPID
: CLEAR PID ADDRESS, DON'T NEED TO REWRITE S
: YES.
: ASSUME CALLERS PID
: GET PNAME ADDRESS IF SPECIFIED
```

50	04	AC	D0	00B2	432
		18	13	00B6	433
				00B8	434
51	60	D0	00BE	435	
	0D	13	00C1	436	
50	51	D0	00C3	437	
	00F9	30	00C6	438	
51	50	D0	00C9	439	
	50	D4	00CC	440	
	65	11	00CE	441	
51	60	A4	D0	00D0	442
53	08	AC	D0	00D4	443

```
5B 13 00D8 444 BEQL GOTPID : NONE SPECIFIED, USE COMMON EXIT
00DA 445 20$: : MUST LOOK UP PROCESS NAME
00DA 446 : CHECK DESCRIPTOR FOR READABILITY
52 63 7D 00E0 447 MOVQ #8,(R3),ACCVIO : GET DESCRIPTOR
52 65 B5 00E3 448 TSTW R2 : AND CHECK FOR ZERO LENGTH
52 41 13 00E5 449 BEQL IVLNAM : NOT A VALID NAME STRING
52 0F B1 00E7 450 CMPW #15,R2 : CHECK FOR MAXIMUM LENGTH
3C 1F 00EA 451 BLSSU IVLNAM : NOT A VALID NAME STRING
00EC 452 IFNORD R2,(R3),ACCVIO : ACCESS VIOLATION IF STRING NOT READABLE
50 DD 00F2 453 PUSHL R0 : SAVE PID ADDRESS
50 00000000'EF D0 00F4 454 MOVL SCH$GL_MAXPIX,R0 : INITIALIZE PROCESS INDEX
51 00000000'FF40 D0 00FB 455 PIXLOOP: : LOOP FOR EACH PROCESS INDEX
00BE C4 00BE C1 B1 0103 456 MOVL @L^SCH$GL_PCBVEC[R0],R1 : GET PCB ADDRESS FROM VECTOR
11 12 010A 457 CMPW PCBSW_GRP(R1),PCBSW_GRP(R4) : COMPARE GROUP NUMBERS
70 A1 52 91 010C 458 BNEQ NEXTPIX : NOT SAME GROUP, NEXT PIX
0B 12 0110 459 CMPB R2,PCBST_LNAME(R1) : COMPARE NAME LENGTH
71 A1 63 0F BB 0112 460 BNEQ NEXTPIX : DIFFERENT LENGTH
52 29 0114 461 PUSHR #*M<R0,R1,R2,R3> : SAVE REGISTERS FOR CMPC3
0F BA 0119 462 CMPC3 R2,(R3),PCBST_LNAME+1(R1) : COMPARE TEXT OF NAME
11 13 011B 463 POPR #*M<R0,R1,R2,R3> : RESTORE REGISTERS
011D 464 BEQL GOTNAM : FOUND A MATCHING PROCESS NAME
DB 50 F4 011D 465 NEXTPIX: : STEP TO NEXT PROCESS
8E D5 0120 466 SOBGEQ R0,PIXLOOP : UPDATE INDEX AND TRY AGAIN
2E 11 0122 467 TSTL (SP)+ : CLEAN PID ADDRESS FROM STACK
0124 468 BRB NONEX : EXIT WITH NONEXISTENT PROCESS STATUS
0124 469
50 0C 3C 0124 470 ACCVIO: : ACCESS VIOLATION
05 0127 471 MOVZWL #SS$_ACCVIO,R0 : SET ERROR CODE
0128 472 RSB : AND EXIT
0128 473
50 0154 8F 3C 0128 474 IVLNAM: : INVALID NAME
05 0128 475 MOVZWL #SS$_IVLOGNAM,R0 : SET ERROR CODE
012D 476 RSB : AND RETURN
012E 477
51 60 A1 D0 012E 478 GOTNAM: MOVL PCBSL_PID(R1),R1 : GET FULL PID FOR NAME
50 BED0 0132 479 POPL R0 : RESTORE PID ADDRESS
0135 480 GOTPID: : VERIFY PID AND CHECK PRIV
0135 481 SETIPL #IPL$_SYNCH : BLOCK SYSTEM EVENTS
52 51 3C 0138 482 MOVZWL R1,R2 : EXTRACT PROCESS INDEX
00000000'EF 52 D1 0138 483 CMPL R2,SCH$GL_MAXPIX : TEST AGAINST MAXIMUM VALUE
0E 1A 0142 484 BGTRU NONEX : NONEXISTENT IF GTRU THAN MAXPIX
52 00000000'FF42 D0 0144 485 MOVL @L^SCH$GL_PCBVEC[R2],R2 : GET PCB ADDRESS
60 A2 51 D1 014C 486 CMPL R1,PCBSL_PID(R2) : CHECK FOR VALID PID
06 13 0150 487 BEQL VALPID : YES
0152 488 NONEX: : PROCESS NON-EXISTENT
0152 489 MOVZWL #SS$_NONEXPR,R0 : SET ERROR STATUS
0157 490 RSB : AND RETURN TO CALLER
0158 491 VALPID: : PID IS VALID, CHECK PRIV
0158 492 CMPL PCBSL_JIB(R2),PCBSL_JIB(R4) : IS IT IN OUR JOB (TREE)?
015F 493 BEQL RETURN : IF SO, ALLOW IT WITHOUT PRIVILEGES
00BC C4 00BC C2 D1 0161 494 CMPL PCBSL_UIC(R2),PCBSL_UIC(R4) : DOES PROCESS HAVE SAME UIC?
15 13 0168 495 BEQL RETURN : IF SO, ALLOW IT WITHOUT PRIVILEGES
016A 496 IFPRIV WORLD,RETURN,R4 : SUCCESS IF WORLD PRIVILEGE
00BE C4 00BE C2 B1 0170 497 CMPW PCBSW_GRP(R2),PCBSW_GRP(R4) : ARE GROUP NUMBERS EQUAL
1C 12 0177 498 BNEQ NOPRIV : IF NOT, NO PRIVILEGE
0179 499 IFNPRIV GROUP,NOPRIV,R4 : ERROR IF NOT GROUP PRIV
017F 500 RETURN: : SUCCESSFUL EXIT
```

54	52	D0	017F	501	MOVL	R2,R4	:	MOVE PCB ADDRESS OF TARGET
			0182	502			:	NORMAL STATUS EXIT
	50	D5	0182	503	TSTL	R0	:	WAS PID ADDRESS SPECIFIED
	08	13	0184	504	BEQL	10\$:	NO, SKIP STORE OF PID
			0186	505	SETIPL	#IPL\$_ASTDEL	:	ALLOW PAGE FAULTS
60	64	A4	D0	0189	MOVL	PCBSL_EPID(R4),(R0)	:	STORE EXTENDED PID IN DESTINATION
	50	D4	018D	507	CLRL	R0	:	DO NOT WRITE PID A SECOND TIME
	A4	11	018F	508	BRB	GOTPID	:	MAKE SURE THAT PID IS STILL VALID
			0191	509			:	
50	01	3C	0191	510	10\$:	MOVZWL	:	SET SUCCESS STATUS
		05	0194	511	RSB	#SS\$ _NORMAL,R0	:	AND RETURN TO CALLER
50	24	3C	0195	512	NOPRIV:	MOVZWL	:	SET ERROR STATUS
		05	0198	513	RSB	#SS\$ _NOPRIV,R0	:	AND RETURN TO CALLER
			0199	514			:	

```
0199 516 .SBTTL EXESPID_TO_XXX - CONVERT PID TO OTHER PID OR PCB ADDRESS
0199 517 :++
0199 518 : FUNCTIONAL DESCRIPTIONS:
0199 519 :
0199 520 :     EXESPID_TO_PCB          - convert internal pid to pcb address
0199 521 :     EXESEPID_TO_PCB        - convert extended pid to pcb address
0199 522 :     EXESPID_TO_EPID        - convert internal pid to extended pid
0199 523 :     EXESEPID_TO_IPID       - convert extended pid to internal pid
0199 524 :
0199 525 : CALLING SEQUENCE:
0199 526 :     JSB/BSB EXESPID_TO_XXX
0199 527 :
0199 528 : INPUT PARAMETERS:
0199 529 :     R0          - input pid
0199 530 :
0199 531 : IMPLICIT INPUTS:
0199 532 :     @SCH$GL_PCBVEC - VECTOR OF PCB ADDRESSES
0199 533 :     SCH$GL_PIXWIDTH - WIDTH OF PIX FIELD IN EXTENDED PID
0199 534 :
0199 535 : OUTPUT PARAMETERS:
0199 536 :     R0          - output pid or pcb address, 0 if any problems
0199 537 :     CONDITION CODES - set according to the value in R0, so that any call
0199 538 :                     can be followed by a BEQL without another test
0199 539 :
0199 540 : COMPLETION CODES:
0199 541 :     NONE
0199 542 :
0199 543 : SIDE EFFECTS:
0199 544 :
0199 545 :     Non-paged code and data, no page faults possible.
0199 546 :
0199 547 :     Callers of these routines must be prepared for the routines to save
0199 548 :     registers R1 through R5 to allow for future additions. For example,
0199 549 :     a BLISS linkage declaration of
0199 550 :
0199 551 :         LINKAGE
0199 552 :             pid_call = JSB (REGISTER=0) : PRESERVE (1,2,3,4,5)
0199 553 :                                     NOTUSED (6,7,8,9,10,11);
0199 554 :
0199 555 :     will force the enclosing procedure to save R2-R5 in the procedure
0199 556 :     entry mask.
0199 557 : --
0199 558 :
0199 559 : +
0199 560 : Convert an extended PID to a PCB address. We will first convert the EPID to an
0199 561 : IPID, then convert the IPID to the PCB address. The condition codes will be set
0199 562 : according to the value in R0.
0199 563 : -
0199 564 : EXESEPID TO PCB:: ; CONVERT EXTENDED PID TO PCB ADDRESS
0199 565 :     BSBB EXESEPID_TO_IPID ; GET THE IPID IN R0
0199 566 :     BEQL 10$ ; COULDN'T CONVERT THE EPID
0199 567 :     BSBB EXESPID_TO_PCB ; CONVERT THE IPID TO THE PCB ADDR
0199 568 : 10$: RSB
0199 569 :
0199 570 : +
0199 571 : Convert internal PID to PCB address. Return 0 if the input IPID does not match
0199 572 : the IPID stored in the corresponding PCB. Set the condition codes according to
```

```
27 10
02 13
01 10
    05
```

```
01A0 573 : the presence of a returned address in R0, so that the BSBx can be followed by a
01A0 574 : BEQL or BNEQ
01A0 575 :
01A0 576 EXESIPID TO PCB:: : CONVERT INTERNAL PID TO PCB ADDRESS
01A0 577 CMPW R0,SCH$GL_MAXPIX : TEST AGAINST MAXIMUM VALUE
01A0 578 BGTRU 10$ : NONEXISTENT IF GTRU THAN MAXPIX
01A0 579 PUSHL R0 : SAVE A COPY OF THE IPID
01AB 580 MOVZWL R0,R0 : EXTRACT PROCESS INDEX FIELD
01AE 581 MOVL @SCH$GL_PCBVEC[R0],R0 : MOVE PCB ADDRESS TO R0
01B6 582 CMPL PCB$P_PID(R0),(SP)+ : DOES THE PID IN THE PCB MATCH?
01BA 583 BNEQ 10$ : NO MATCH, RETURN 0 ADDRESS
01BC 584 TSTL R0 : SET THE CONDITION CODES
01BE 585 RSB
01BF 586 10$: CLRL R0 : NONEXISTENT PID, RETURN ZERO
01C1 587 RSB
01C2 588
01C2 589 :+
01C2 590 : Convert an extended PID to the internal PID. Return 0 if the EPID refers to
01C2 591 : another node. Do not check that either the EPID or IPID are valid.
01C2 592 :
01C2 593 EXESEPID TO IPID:: : CONVERT EXTENDED PID TO INTERNAL PID
01C2 594 PUSHR #*M<R1,R2> : SAVE SOME WORKING REGISTERS
01C4 595
01C4 596 : WE WILL EXTRACT THE NODE FIELD FROM THE EPID TO SEE IF THIS IS FOR THE LOCAL
01C4 597 : NODE. WE WILL INCLUDE THE WILDCARD BIT IN THIS TEST. VERIFY SOME ASSUMPTIONS
01C4 598 : ABOUT THE LOCATIONS OF THESE FIELDS.
01C4 599
01C4 600 NODE_WIDTH = PCB$S_EPID_NODE_IDX+PCB$S_EPID_NODE_SEQ
01C4 601
01C4 602 ASSUME PCB$V_EPID_WILD EQ - : CHECK THAT WILD BIT IS RIGHT
01C4 603 <PCB$V_EPID_NODE_IDX + NODE_WIDTH> : AFTER NODE FIELDS
01C4 604 ASSUME PCB$V_EPID_NODE_SEQ EQ - : AND SEQ IS RIGHT AFTER IDX
01C4 605 <PCB$V_EPID_NODE_IDX + PCB$S_EPID_NODE_IDX>
01C4 606
01C4 607 EXTZV #PCB$V_EPID_NODE_IDX, - : MOVE NODE + WILD TO R1
01C9 608 #<NODE_WIDTH+1>,R0,R1
01C9 609 BEQL 10$ : TREAT NODE ZERO AS LOCAL NODE ??
01CB 610 CMPW SCH$GL_LOCALNODE,R1 : IS IT THE LOCAL NODE?
01D2 611 BNEQ 30$ : NOT LOCAL, CAN'T MAKE AN IPID
01D4 612
01D4 613 : EPID IN R0 IS FOR LOCAL NODE, EXTRACT THE PIX AND SEQUENCE NUMBER TO FORM IPID
01D4 614
01D4 615 10$: MOVL SCH$GL_PIXWIDTH,R1 : LOAD WIDTH OF EXTENDED PIX FIELD
01DB 616 SUBL3 R1,#PCB$S_EPID_PROC,R2 : AND WIDTH OF THE SEQ NUM FIELD
01DF 617 EXTZV R1,R2,R0,R2 : R2 IS LONGWORD SEQ NUM
01E4 618 EXTZV #0,R1,R0,R0 : R0 IS LONGWORD PIX
01E9 619 INSV R2,#16,#15,R0 : INSERT SEQ NUM IN HIGH WORD
01EE 620 : WHICH MAKES AN IPID IN R0
01EE 621 20$: POPR #*M<R1,R2> : RESTORE REGISTERS
01F0 622 RSB : CONDITION CODES SET FOR VALUE OF R0
01F1 623
01F1 624 : COULD NOT TURN EPID INTO AN IPID, RETURN AN IPID OF 0
01F1 625
01F1 626 30$: CLRL R0 : RETURN ZERO PID (& COND CODE = 0)
01F3 627 BRB 20$ : RESTORE REGISTERS AND RETURN
01F5 628
01F5 629 :+
```

EXESxPID_TO_XXX

- CONVERT PID TO OTHER P

01F5 630 : Convert an IPID to an EPID. We do not check that the IPID is valid. The local
01F5 631 : node is moved into the node field of the EPID, the seq number and pix of the IPID
01F5 632 : are moved into the EPID. The condition codes reflect the final value of R0.
01F5 633 :
01F5 634

EXESIPID TO EPID::

; INTERNAL PID TO EXTENDED PID

50	D5	01F5	635	TSTC	R0	:	TREAT A ZERO PID AS A SPECIAL CASE
28	13	01F5	636	BEQL	108	:	ZERO, WE DON'T TOUCH IT
OE	BB	01F7	637	PUSHR	#*M<R1,R2,R3>	:	SAVE SOME WORKING REGISTERS
50	3C	01F9	638	MOVZWL	R0,R3	:	R3 IS LONGWORD PIX
53	78	01FB	639	ASHL	#-16,R0,R0	:	R0 IS LONGWORD SEQ NUM
50	DO	0203	640	MOVL	SCH\$GL_PIXWIDTH,R1	:	LOAD WIDTH OF EXTENDED PIX FIELD
51	C3	020A	641	SUBL3	R1,#PCB\$S_EPID_PROG,R2	:	AND THE WIDTH OF THE SEQ NUM FIELD
52	FO	020E	642	INSV	R0,R1,R2,R3	:	INSERT SEQ NUM BESIDE PIX
53	FO	0213	643	INSV	SCH\$GW_LOCALNODE, -	:	INSERT LOCAL NODE INTO THE EPID
50	DO	021C	644		#PCB\$V_EPID_NODE_IDX, #NODE_WIDTH, R3	:	
53	BA	021F	645	MOVL	R3,R0	:	RETURN THE EPID IN R0
OE	OS	0221	646	POPR	#*M<R1,R2,R3>	:	RESTORE REGISTERS
		0222	647	RSB		:	N.B. COND CODES SET ON VALUE OF R0
			648			:	

108:

```
0222 650 .SBTTL EXESSETPRN - SET PROCESS NAME
0222 651
0222 652 :++
0222 653 : FUNCTIONAL DESCRIPTION:
0222 654 : EXESSETPRN IMPLEMENTS THE SET PROCESS NAME SYSTEM
0222 655 : SERVICE WHICH ALLOWS A PROCESS TO ESTABLISH A LOGICAL NAME
0222 656 : FOR ITSELF. ALL SUCH LOGICAL NAMES ARE IMPLICITLY QUALIFIED
0222 657 : BY THE GROUP NUMBER OF THE PROCESS THEREBY ALLOWING THE SAME
0222 658 : LOGICAL NAME TO BE USED BY PROCESSES IN DIFFERENT GROUPS.
0222 659
0222 660 : CALLING SEQUENCE:
0222 661 : CALLG ARGLIST,EXESSETPRN
0222 662
0222 663 : INPUT PARAMETERS:
0222 664 : 04(AP) - ADDRESS OF PROCESS NAME STRING DESCRIPTOR
00000004 0222 665 : PRCNAM=4
0222 666 : R4 - PCB ADDRESS OF CURRENT PROCESS
0222 667
0222 668 : IMPLICIT INPUTS:
0222 669 : SCH$GL_CURPCB - POINTER TO PCB OF CURRENT PROCESS
0222 670 : @SCH$GL_PCBVEC - VECTOR OF ALL PCB ADDRESSES
0222 671
0222 672 : OUTPUT PARAMETERS:
0222 673 : NONE
0222 674
0222 675 : IMPLICIT OUTPUTS:
0222 676 : PCB$T_NAME IN CURRENT PCB IS FILLED WITH THE SPECIFIED NAME
0222 677 : PROVIDED NO ERROR HAS OCCURRED.
0222 678
0222 679 : SIDE EFFECTS:
0222 680 : NONE
0222 681
0222 682 : COMPLETION CODES:
0222 683 : SSS_NORMAL - NORMAL SUCCESSFUL COMPLETION STATUS
0222 684 : SSS_ACCVIO - ALL OR PART OF NAME STRING IS INACCESSIBLE FOR READ
0222 685 : SSS_IVLOGNAM - ILLEGAL LOGICAL NAME STRING LENGTH (>15)
0222 686 : SSS_DUPLNAM - DUPLICATE PROCESS NAME WITHIN GROUP
0222 687
0222 688 :--
0222 689
0222 690 EXESSETPRN::
0222 691 .WORD ^M<R2,R3,R4,R5,R6,R7> : SET PROCESS NAME
0222 692 : MOVL PRCNAM(AP),R5 : SAVE REGISTERS R2-R7
0222 693 : BNEQ S$ : GET ADDRESS OF PROCESS NAME
0222 694 : CLRL PCB$T_LNAME(R4) : WAS SPECIFIED
0222 695 : BRB 65$ : CLEAR NAME FIELD OF PCB
0222 696 5$: IFNORD #8,(R5),80$ : AND EXIT WITH NORMAL STATUS
0222 697 : MOVQ (R5),-(SP) : CHECK ACCESS FOR DESCRIPTOR
0222 698 : TSTW (SP) : PUSH DESCRIPTOR ON STACK
0222 699 : BEQL 10$ : CHECK FOR ZERO LENGTH STRING
0222 700 : IFNORD (SP),@4(SP),80$ : INVALID NAME
0222 701 : CMPW (SP),#15 : PROBE ENDS OF STRING
0222 702 : BLEQU 20$ : CHECK FOR MAXIMUM LENGTH
0222 703 10$: MOVZWL #SS$_IVLOGNAM,R0 : IF LEQU, WITHIN LIMIT
0222 704 : RET : INVALID PROCESS NAME STATUS
0222 705 20$: MOVL SCH$GL_MAXPIX,R6 : AND RETURN
0222 706 30$: MOVL @L^SCH$GL_PCBVEC[R6],R7 : SET MAXIMUM PROCESS INDEX
0222 : : GET PCB ADDRESS
```

55 04 AC 00FC 0222 691
05 12 0222 692
70 A4 D4 0222 693
59 11 0222 694
7E 65 7D 0222 695
6E B5 0222 696
OC 13 0222 697
OF 6E B1 0222 698
06 1B 0222 699
50 0154 8F 3C 0222 700
04 0222 701
56 00000000'EF D0 0222 702
57 00000000'FF46 D0 0222 703

00BE C7	00BE C4	B1	025D	707	CMPL	PCBSW_GRP(R4),PCBSW_GRP(R7)	; CHECK FOR SAME GROUP
	OE	12	0264	708	BNEQ	40\$; NO, SKIP IT
70 A7	6E	91	0266	709	CMPL	(SP),PCBST_LNAME(R7)	; COMPARE LENGTHS
	08	12	026A	710	BNEQ	40\$; NOT EQUAL, TRY ANOTHER
71 A7	04 BE	29	026C	711	CMPC3	(SP),B4(SP),PCBST_LNAME+1(R7)	; COMPARE NAMES WITH COUNTS
	05	13	0272	712	BEQL	50\$; MATCH
	DE 56	F4	0274	713	SOBGEQ	R6,30\$; CONTINUE FOR ALL PCBS
	05	11	0277	714	BRB	60\$; NOT FOUND
	57	D1	0279	715	CMPL	R4,R7	; SAME PROCESS?
	OE	12	027C	716	BNEQ	70\$; DUPLICATE NAME ERROR
71 A4	70 A4	90	027E	717	MOVB	(SP),PCBST_LNAME(R4)	; SAVE NAME LENGTH
	04 BE	28	0282	718	MOV3	(SP),B4(SPT),PCBST_LNAME+1(R4)	; MOVE NAME TO PCB
	50	3C	0288	719	MOVZWL	#SS\$_NORMAL,R0	; SUCCESSFUL STATUS
	01	04	028B	720	RET		; AND RETURN
50	0094 8F	3C	028C	721	MOVZWL	#SS\$_DUPLNAM,R0	; DUPLICATE NAME WITHIN GROUP
		04	0291	722	RET		; AND RETURN
			0292	723			
	50	3C	0292	724	MOVZWL	#SS\$_ACCVIO,R0	; ACCESS VIOLATION
	OC	04	0295	725	RET		; RETURN WITH ERROR STATUS
			0296	726	.END		

Symbol	Value	Mode	Value
ACBSB_RMOD	=	0000000B	
ACBSL_AST	=	00000010	
ACBSL_PID	=	0000000C	
ACCVID		00000124	R 02
EVT\$ RESUME	*****		X 02
EXES\$ALLOCIRP	*****		X 02
EXESEPID_TO_IPID		000001C2	RG 02
EXESEPID_TO_PCB		00000199	RG 02
EXES\$HIBER		0000008B	RG 02
EXESEPID_TO_EPID		000001F5	RG 02
EXESEPID_TO_PCB		000001A0	RG 02
EXES\$NAMPID		000000B2	RG 02
EXES\$RESUME		00000076	RG 02
EXES\$SETPRN		00000222	RG 02
EXES\$SUSPND		00000000	RG 02
EXES\$WAKE		000000A1	RG 02
EXIT		000000AE	R 02
EXITN		000000AB	R 02
EXIT_NO_POOL		0000004D	R 02
GOTNAM		0000012E	R 02
GOTPID		00000135	R 02
IPL\$ ASTDEL	=	00000002	
IPL\$ SYNCH	=	00000008	
IVLNAM		00000128	R 02
NEXTPIX		0000011D	R 02
NODE_WIDTH	=	0000000A	
NONEX		00000152	R 02
NOPRIV		00000195	R 02
PCBSB_ASTACT	=	0000000C	
PCBSB_DPC	=	0000002A	
PCBSL_EPID	=	00000064	
PCBSL_JIB	=	00000080	
PCBSL_PID	=	00000060	
PCBSL_STS	=	00000024	
PCBSL_UIC	=	000000BC	
PCBSQ_PRIV	=	00000084	
PCBS\$ EPID_NODE_IDX	=	00000008	
PCBS\$ EPID_NODE_SEQ	=	00000002	
PCBS\$ EPID_PROC	=	00000015	
PCBST_LNAME	=	00000070	
PCBSV_DELPEN	=	00000001	
PCBSV_EPID_NODE_IDX	=	00000015	
PCBSV_EPID_NODE_SEQ	=	0000001D	
PCBSV_EPID_WILD	=	0000001F	
PCBSV_RESPEN	=	00000005	
PCBSV_SUSPEN	=	0000000B	
PCBSV_WAKEPEN	=	0000000C	
PCBSW_GRP	=	000000BE	
PID	=	00000004	
PIXLOOP		000000FB	R 02
PR\$ IPL	=	00000012	
PR\$NAM	=	00000004	
PR\$ RESAVL	=	00000002	
PRV\$V_GROUP	=	00000008	
PRV\$V_WORLD	=	00000010	
RETURN		0000017F	R 02
RSNS_ASTWAIT	=	00000001	

SCH\$GL_CURPCB	*****	X	02
SCH\$GL_MAXPIX	*****	X	02
SCH\$GL_PCBVEC	*****	X	02
SCH\$GL_PIXWIDTH	*****	X	02
SCH\$GQ_HIBWQ	*****	X	02
SCH\$GQ_SUSP	*****	X	02
SCH\$GW_LOCALNODE	*****	X	02
SCH\$NEWLVL	*****	X	02
SCH\$QAST	*****	X	02
SCH\$RSE	*****	X	02
SCH\$RWAIT	*****	X	02
SCH\$WAIT	*****	X	02
SCH\$WAITK	*****	X	02
SCH\$WAKE	*****	X	02
SS\$_ACCVIO	= 0000000C		
SS\$_DUPLNAM	= 00000094		
SS\$_IVLOGNAM	= 00000154		
SS\$_NONEXPR	= 000008E8		
SS\$_NOPRIV	= 00000024		
SS\$_NORMAL	= 00000001		
SUSPND	0000003A	R	02
VALPID	00000158	R	02

SYS
Sym

BUG
BUG
EXE
INA
IPL
IPL
MMG
MMG
MMG
MMG
MMG
MMG
MMG
MMG
MMG
MMG
PCB
PCB
PFN
PHD
PHD
PHD
PHD
PRS
PSL
PSL
PTE
PTE
PUR
SS\$
SS\$
WSL
WSL
WSL

PSE

SAB
YSE
SHNPha

Int
Com
Pas

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
AEXENONPAGED	00000296 (662.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.07	00:00:00.29
Command processing	105	00:00:00.56	00:00:01.82
Pass 1	288	00:00:08.52	00:00:16.46
Symbol table sort	0	00:00:01.25	00:00:02.88
Pass 2	141	00:00:02.21	00:00:05.49
Symbol table output	11	00:00:00.07	00:00:00.14
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	578	00:00:12.70	00:00:27.10

The working set limit was 1500 pages.
49110 bytes (96 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 807 non-local and 28 local symbols.
726 source lines were read in Pass 1, producing 16 object records in Pass 2.
23 pages of virtual memory were used to define 22 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	12
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	7
TOTALS (all libraries)	19

909 GETS were required to define 19 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SYSPCNTRL/OBJ=OBJ\$:SYSPCNTRL MSRC\$:SYSPCNTRL/UPDATE=(ENH\$:SYSPCNTRL)+EXECML\$/LIB

0387

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY